

z/OS



MVS System Commands Summary

z/OS



MVS System Commands Summary

Note

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 59.

Third Edition, September 2002

This is a major revision of SA22-7628-02.

This edition applies to Version 1 Release 4 of z/OS (5694-A01), Version 1 Release 4 of z/OS.e (5655-G52), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order documents through your IBM® representative or the IBM branch office serving your locality. Documents are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405

FAX (Other Countries):

Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)

Internet e-mail: mhvrcfs@us.ibm.com

World Wide Web: <http://www.ibm.com/servers/eserver/zseries/zos/webqs.html>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1988, 2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this document v

Who should use this document v

Information updates on the web. v

Using LookAt to look up message explanations . . . v

Accessing z/OS™ licensed documents on the

Internet. vi

Summary of changes vii

How to read syntax conventions ix

MVS System Commands Syntax 1

ACTIVATE Command 1

CANCEL Command (C) 1

CHNGDUMP Command (CD) 1

CMD5 Command. 3

CONFIG Command (CF) 3

CONTROL Command (K) 4

DEVSERV Command (DS). 7

DISPLAY Command (D) 8

DUMP Command 21

DUMPDS Command (DD) 21

FORCE Command 22

HALT Command (Z) 23

IOACTION Command (IO) 23

LIBRARY Command (LI) 23

LOG Command (L). 24

LOGOFF Command 24

LOGON Command. 24

MODE Command 24

MODIFY Command (F) 25

MONITOR Command (MN). 30

MOUNT Command (M) 30

MSGRT Command (MR) 30

PAGEADD Command (PA) 31

PAGEDEL Command (PD) 31

QUIESCE Command 31

REPLY Command (R) 31

RESET Command (E) 33

ROUTE Command (RO) 34

SEND Command (SE) 34

SET Command (T) 35

SETDMN Command (SD) 36

SETETR Command. 36

SETGRS Command. 36

SETIOS Command 36

SETLOAD Command 37

SETLOGRC Command 37

SETOMVS Command 37

SETPROG Command 38

SETRRS CANCEL Command 40

SETSMF Command (SS) 40

SETSMS Command. 40

SETSSI Command 40

SETXCF Command. 41

SLIP Command (SL) 43

START Command (S) 47

STOP Command (P) 48

STOPMN Command (PM) 49

STOPTR Command (PT) 50

SWAP Command (G) 50

SWITCH Command (I) 50

TRACE Command 50

TRACK Command (TR) 51

UNLOAD Command (U). 51

VARY Command (V) 52

WRITELOG Command (W) 55

Appendix. Accessibility 57

Using assistive technologies 57

Keyboard navigation of the user interface 57

Notices 59

Trademarks 60

About this document

This document states the purpose and shows the syntax of the MVS system commands (also known as "operator commands" or "console commands") you can use to control processors under the z/OS and z/OS.e operating systems. For the full explanations and detailed descriptions of the MVS system command functions, syntax, and parameters, see *z/OS MVS System Commands*.

This document includes only the MVS *system* commands, which it lists in alphabetical order. For information about JES subsystem commands and their functions, see *z/OS JES2 Commands* or *z/OS JES3 Commands*.

For information about the system messages referred to in this document, see *z/OS MVS System Messages*.

Who should use this document

z/OS MVS System Commands Summary is intended for anyone using a console and operator commands to manage the z/OS operating system. It assumes that the user understands the hardware controls and features of the installation and the general organization and functions of the z/OS operating system.

Information updates on the web

For the latest information updates that have been provided in PTF cover letters and Documentation APARs for z/OS and z/OS.e, see the online document at:

<http://www.s390.ibm.com:80/bookmgr-cgi/bookmgr.cmd/BOOKS/ZIDOCMST/CCONTENTS>

This document is updated weekly and lists documentation changes before they are incorporated into z/OS publications.

Using LookAt to look up message explanations

LookAt is an online facility that allows you to look up explanations for most messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can access LookAt from the Internet at:

<http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/>

or from anywhere in z/OS where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX System Services running OMVS). You can also download code from the *z/OS Collection* (SK3T-4269) and the LookAt Web site that will allow you to access LookAt from a handheld computer (Palm Pilot VIIx suggested).

To use LookAt as a TSO/E command, you must have LookAt installed on your host system. You can obtain the LookAt code for TSO/E from a disk on your *z/OS Collection* (SK3T-4269) or from the **News** section on the LookAt Web site.

Some messages have information in more than one document. For those messages, LookAt displays a list of documents in which the message appears.

Accessing z/OS™ licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at:

<http://www.ibm.com/servers/resourceLink>

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.¹

To obtain your IBM Resource Link user ID and password, log on to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

1. z/OS.e™ customers received a Memo to Licensees, (GI10-0684) that includes this key code.

Summary of changes

Summary of changes for SA22-7628-03 z/OS Version 1 Release 4

This document contains information previously presented in *z/OS MVS System Commands Summary*, SA22-7628-02, which supports z/OS Version 1 Release 3.

New information

- Text is added to indicate this document supports z/OS.e.
- Two new parameters, AUTHPGMLIST and AUTOMOVE, have been added for the SETOMVS command.
- MODIFY CATALOG,NOTIFYEXTENT(xxx) is a new command that allows you to specify the maximum extents possible for a catalog that are currently allocated.

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Starting with z/OS V1R2, you may notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

Summary of changes for SA22-7628-02 z/OS Version 1 Release 3

This document contains information previously presented in *z/OS MVS System Commands Summary*, SA22-7628-01, which supports z/OS Version 1 Release 2.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

Summary of changes for SA22-7628-01 z/OS Version 1 Release 2

This document contains information previously presented in *z/OS MVS System Commands Summary*, SA22-7628-00, which supports z/OS Version 1 Release 1.

New Information

- The CMDS command is added.
- The ACTION=STOPGTF and MSGID= parameters are added to the SLIP command.

Deleted Information

- The SLIP SET,SB1 parameter is removed. That parameter is supported only on processors that z/OS no longer supports.

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability.

**Summary of changes
for SA22-7628-00
z/OS Version 1 Release 1**

This document contains information also presented in *OS/390 MVS System Commands Summary*.

How to read syntax conventions

This section describes how to read syntax conventions. It defines syntax notations and provides syntax examples that contain these items.

Table 1. Syntax conventions

Notation	Meaning	Example	
		Book Syntax	Sample Entry
Apostrophes	Apostrophes indicate a parameter string and must be entered as shown.	SEND 'message',NOW	SEND 'listings ready',NOW
Comma	Commas must be entered as shown.	DISPLAY C,K	DISPLAY C,K
Ellipsis ...	Ellipsis indicates that the preceding item or group of items can be repeated one or more times. Do not enter the ellipsis.	VARY (devspec[,devspec]...),ONLINE	VARY (282,283,287),ONLINE
Parentheses and special characters	Parentheses and special characters must be entered as shown.	DUMP COMM=(text)	DUMP COMM=(PAYROLL)
Underline	Underline indicates a default option. If you select an underlined alternative, you do not have to specify it when you enter the command.	K T [<u>,REF</u>] [<u>,UTME=nnn</u>]	K T
Lowercase parameter	Lowercase indicates a variable term. Substitute your own value for the item.	MOUNT devnum	MOUNT A30 or mount a30
Uppercase parameter	Uppercase indicates the item must be entered using the characters shown. Enter the item in either upper or lowercase.	DISPLAY SMF	DISPLAY SMF or display smf
Single brackets	Single brackets represent single or group-related items that are optional. Enter one or none of these items.	DISPLAY DMN[=domainum]	DISPLAY DMN=5
Stacked brackets	Stacked brackets represent group-related items that are optional. Enter one or none of these items.	[<u>TERMINAL</u>] [<u>NOTERMINAL</u>]	NOTERMINAL

Table 1. Syntax conventions (continued)

Notation	Meaning	Example	
		Book Syntax	Sample Entry
Single braces	Single braces represent group-related items that are alternatives. You must enter one of the items. You cannot enter more than one.	{COMCHECK COMK}	COMK
Stacked braces	Stacked braces represent group related items that are alternatives. You must enter one of the items. You cannot enter more than one.	MN {DSNAME} {SPACE } {STATUS}	MN SPACE
Or-bar ()	An or-bar indicates a mutually-exclusive choice. When used with brackets, enter one or none of the items. When used with braces, you must enter one of the items.	ACTIVATE RECOVER=SOURCE	RECOVER=SOURCE
Stacked items with or-bars () and brackets	Stacked items with or-bars indicates a mutually-exclusive choice. Enter one or none of these items.	CD RESET [,SDUMP] ,SYSABEND ,SYSUDUMP ,SYSMDUMP ,ALL	CD RESET,SYSUDUMP

MVS System Commands Syntax

ACTIVATE Command

Purpose: Use the ACTIVATE command to activate or test a new I/O configuration definition dynamically.

ACTIVATE
<pre>ACTIVATE { [,IODF=xx] [,EDT=xx] [,PROC=procname] [,CFID=id] } [,RECOVER=SOURCE TARGET] [,ACTIOCDS=xx] { [,SOFT[=VALIDATE =NOVALIDATE]] } { ,TEST } { ,FORCE } { ,FORCE={DEVICE } {CANDIDATE } {(DEVICE,CANDIDATE)} {(CANDIDATE,DEVICE)} }</pre>

Note: Do not specify a comma before the first parameter following ACTIVATE.

CANCEL Command (C)

Purpose: Use the CANCEL command to end an active job, started task, or time-sharing user immediately.

CANCEL or C
<pre>C {jobname } [,DUMP] [,A=asid] [,ARMRESTART] {U=userid } {[jobname.]identifier}</pre>

CHNGDUMP Command (CD)

Purpose: Use the CHNGDUMP command to change the mode and system dump options list for any dump type, or to request structures to be dumped when one or more systems connected to a coupling facility fail.

Example: The following CHNGDUMP command removes options from the options lists.

CHNGDUMP Command

```

CD DEL[,SDUMP[(option[,option]...)]
           [,Q={YES|NO}]
           [,TYPE={XMEM|XMEME}]
           [,ALL]
           [,SYSFAIL,STRLIST={ALL|(STRNAME=strname[,STRNAME=strname]...)}]
[,{,SYSABEND}[,SDATA=(option[,option]...)]
[,{,SYSUDUMP} |,PDATA=(option[,option]...)]
           |,ALL
[,{,SYSDUMP[(option[,option]...)]
           |,ALL]
[,{,ALL}

```

Example: The following CHNGDUMP command resets the options lists.

```

CD RESET[,SDUMP
           |,SYSABEND
           |,SYSUDUMP
           |,SYSDUMP
           |,ALL]

```

Example: The following CHNGDUMP command sets the dump modes and options.

```

CD SET,{NODUMP
           {OVER
           {ADD
           {SDUMP[(option[,option]...)]
           [,Q={YES|NO}]
           [,TYPE={XMEM|XMEME}]
           [,BUFFERS={nnnnK|nnnM}]
           [,MAXSPACE=xxxxxxxxM]
           [,MSGTIME=yyyyy]
           [,SYSFAIL,STRLIST=(s-option[,s-option]...)]
           [,NODUMP]
           |,OVER
           |,ADD
           {{,SYSABEND}[,SDATA=(option[,option]...)] [,NODUMP] }
           {{,SYSUDUMP} |,PDATA=(option[,option]...)} |,OVER }
           |,ADD
           {,SYSDUMP[(option[,option]...)] [,NODUMP] }
           |,OVER
           |,ADD]

```

Where **s-option** represents:

```

STRNAME=strname
[,CONNAME=conname ]
[,ACCESSTIME={ENFORCE|NOLIMIT|NOLIM} ]
[,LOCKENTRIES ]
[,USERCNTLS ]
[,EVENTQS ]
[, (EMCONTROLS={ALL|(list)}) ]
[, ({COCLASS|STGCLASS|LISTNUM}={ALL|(list)}) ]
[ {[,ADJUNCT={CAPTURE|DIRECTIO}][,ENTRYDATA={UNSERIALIZE|SERIALIZE}} ] ]
[ {[,SUMMARY} ] ]

```

CMDS Command

Purpose: Use the CMDS command to display executing and waiting MVS commands or to delete commands that are waiting for execution.

CMDS

```

CMDS {ABEND,CMD=ccccccc,ID=nnnn[,CLASS=classname][,JOB=jobname]}
      {DISPLAY|D[,CLASS=classname][,CMD=ccccccc][,ID=nnnn][,JOB=jobname]}
      {REMOVE|R[,CLASS=classname][,CMD=ccccccc][,ID=nnnn][,JOB=jobname]}
      {SHOW|S[,CLASS=classname][,CMD=ccccccc][,ID=nnnn][,JOB=jobname]}

```

Notes:

1. The ABEND parameter requires that you specify the CMD= and ID= subparameters.
2. The REMOVE parameter requires that you specify at least one subparameter.
3. You may specify the optional subparameters of the default or specified parameter in any order.

CONFIG Command (CF)

Purpose: Use the CONFIG command to change or check the configuration of the system.

Example: The following CONFIG command reconfigures the system (available processors, storage sections, vector facilities, and channel paths) directly.

CONFIG Command

```
CF {{CPUAD|CPU}(x[,x]...)[,{ONLINE|ON}[,{VFON }|,{OFFLINE|OFF}]}
    |,VFOFF
    {VF(x[,x]...)[,{ONLINE|ON]|,{OFFLINE|OFF}}
    {{STORAGE|STOR}{{(dddM)}[,{ONLINE|ON]|,{OFFLINE|OFF}}
    {{(dddM-dddM)}
    {{(E=id)}
    {ESTOR(E=id)[,{ONLINE|ON]|,{OFFLINE|OFF}}
    {CHP{(xx)}[,{ONLINE|ON][NOVARY]|,{OFFLINE|OFF}[,{UNCOND}]}
    {(aa-bb)}|,FORCE
    {(list)}
    {(ALL,id)}
```

Example: The following CONFIG command reconfigures the system with a CONFIGxx member of SYS1.PARMLIB.

```
CF MEMBER[(member-id)]
```

Example: The following CONFIG command displays the system configuration so that you can decide which processors, central or expanded storage elements, vector facilities, and channel paths you want to bring online/offline. It then brings online/offline items specified in response to message IEE522D.

```
CF {ONLINE|ON}[,{L={a|cc|cca|name|name-a}]}
    {OFFLINE|OFF}
```

CONTROL Command (K)

Purpose: Use the CONTROL command to control the screen display of MCS consoles.

Example: The following CONTROL command changes the display area specifications.

```
K A[,nn[,nn]...][,{L={cc }}
    |,NONE {name}
    |,REF
```

Example: The following CONTROL command deletes one or more action messages that the action message retention facility has retained.

```
K C,{A|I|E|CE},{id|id-id[,id|id-id]...}
```


CONTROL Command

Example: The following CONTROL command stops an in-line status display.

```
K C,D,id[,L={a|cc|cca|name|name-a}]
```

Example: The following CONTROL command displays information on the screen.

```
K D[,N[,HOLD]
  |{,F|,H|,U}[,L={a|cca|name-a}]
  |,PFK
```

Example: The following CONTROL command removes a single deletable message from the screen.

```
K E[,nn
  |,nn,nn
  |,SEG
  |,F
  |,N
  |,PFK
  |,D[,L={a|cc|cca|name|name-a}]
```

Example: The following CONTROL commands activate (Y) or deactivate (N) the action message retention facility.

```
K M[,AMRF={Y|N}]
  |,REF
```

```
K M[,REF|[,MLIM=nnnn][,RLIM=mmmm]]
```

```
K M[,UEXIT={Y|N}]
  |,REF
```

Example: The following CONTROL command displays the maximum number of write-to-log requests that the system can hold in buffers.

```
K M[,LOGLIM={nnnnn|0}]
  |,REF
```

CONTROL Command

Example: The following CONTROL command displays or changes the maximum time that ROUTE commands wait for a response before aggregating responses.

```
K M[,ROUTTIME=nnn ]  
  |,REF
```

Example: The following CONTROL command dynamically increases the maximum number of reply IDs.

```
K M[,RMAX=nnnn ]  
  |,REF
```

Example: The following CONTROL command defines the program function keys.

```
K N,PFK={{(nn1{,CMD='text[;text]...'})[,CON={Y|N}] }  
         { ,KEY=nn2[,nn2]... }  
         {  
         {nnnnnnnn[,L={cc|name}] }  
         }
```

Example: The following CONTROL command changes or displays a console's message queue.

```
K Q[,R={dd }][,L={cc|name}]  
  {name}  
  {HC }
```

Example: The following CONTROL command changes or displays message deletion and format specifications.

```
K S[,REF  
  |[,CON={Y|N}][,SEG=nn][,DEL={Y|N|R|RD|W}]  
  [,RNUM=nn][,RTME=nnn][,MFORM=(option[,option]...)]  
  [,L={cc|name}]
```

Example: The following CONTROL command changes or displays time intervals for dynamic displays.

```
K T[,REF  
  |,UTME=nnn }][,L={cc }  
  {name}
```

Example: The following CONTROL commands change the operating mode or message levels of a console.

```

K V[,REF                                ][,L={cc }]]
    |                                     {name}
    |,USE={FC|SD|MS}[,CMDSYS={sysname|*}]

```

```

K V[,REF                                ][,L={cc }]]
    |                                     {name}
    |,LEVEL=(type[,type]...)

```

DEVSERV Command (DS)

Purpose: Use the DEVSERV command to request a display of the status of DASD and tape devices.

DEVSERV or DS

```

DS {PATHS|P},[/]devnum[,nn][,ONLINE|ON ][,NOSYM|NOS][,DUMP]
  {SMS|S } |,OFFLINE|OFF

    [,L={a|cc|cca}]

  {QDASD | QD}[,? |
    [ [ [,ccuu [,1] ] |,VOL=volser ] [ [,UCB] [,DCE][,SSSCB] [,DPCT]
      [ [,NOIO] ] |[,RDC] [,RCD] [,SNSS] ] ] ] | ,ccuu,nnn |
      [,ccuu,nnn] ,VOL=volser |
        ,MACH=[mmp-sssss | XXXX-sssss] |
        ,SSID=[ssid | ALL] |
        ,TYPE=[type | ALL] ]
    [,ONLINE] [,OFFLINE] [,DEFINED] [,CHKFAIL] [,VALIDATE] [,TOTALCYL]

  {QTAPE | QT}[,? |
    [ [ [,ccuu [,1] ] [ [,UCB] [,DCE][ [,NOIO] | [,RDC]
      [,RCD] ] ] ] | ,ccuu,nnn |
      [,ccuu,nnn] ,LIB=libid | ALL |
        ,MACH=[mmp-sssss | XXXX-sssss] |
        ,TYPE=[type | ALL] ]
    [,ONLINE] [,OFFLINE] [,DEFINED]

  {QPAVS | QP},{dddd}
    {dddd,nn}
    {SSID=ssid}
    {dddd,{VOLUME | UCB | UNBOX}}

```

DISPLAY Command

DISPLAY Command (D)

Purpose: Use the DISPLAY command to display information about the operating system, the jobs and application programs that are running, the processor, devices that are online and offline, central and expanded storage, workload management service policy and mode status, and the time of day.

Example: The following DISPLAY command shows APPC configuration information.

```
D APPC,{TP[ ,SUMMARY|SUM|S ][, {ASID|A}=asid] }
      |,LIST|,L      [,ASNAME=asname]
      |,ALL|,A      [,DIR=IN|OUT]
                        [,IT=sssss[.ttt]]
                        [,LLUN=lluname]
                        [,LTPN=ltpname]
                        [,PNET=pnetid]
                        [,PLUN=pluname]
                        [,PTPN=ptpname]
                        [,SCHED={schedname}]
                        { *NONE* }
                        [,STPN=stpname]
                        [,USERID=userid]

{UR[ ,SUMMARY|SUM|S ][,URID=urid] }
  |,LIST|,L      [,LUWID=luid]
  |,ALL|,A      [,PNET=pnetid]
                [,PLUN=pluname]
                [,LLUN=lluname]

{SERVER[ ,SUMMARY|SUM|S ][, {ASID|A}=asid] }
  |,LIST|,L      [,ASNAME=asname]
  |,ALL|,A      [,LLUN=lluname]
                [,STPN=stpname]

{LU[ ,SUMMARY|SUM|S ][,LLUN=lluname] }
  |,LIST|L      [,PNET=pnetid]
                [,PLUN=pluname]
  |,ALL|A      [,SCHED={schedname}]
                { *NONE* }

[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows ASCH configuration information.

```
D ASCH{ [,SUMMARY|SUM|S ][, {ASID|A}=asid] }
  { |,LIST|,L      [, {CLASS|C}=classname] }
  { |,ALL|,A      [,LTPN=ltpname] }
  {               [,QT=sssss[.ttt]] }
  {               [,TPST=schedtype] }
  {               [,USERID=userid] }

[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows page data set information.

DISPLAY Command

```
D ASM[,PLPA                ][,L={a|cc|cca|name}name-a}]
      |,COMMON
      |,LOCAL
      |,ALL
      |,PAGE=[dsname|ALL]
      |,PAGEDEL
```

Example: The following DISPLAY command shows a summary of CONTROL command operands.

```
D C,K[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows information about the coupling facilities that are attached to the system.

```
D CF[,CFNAME={ (cfname[,cfname]...)}]
```

Example: The following DISPLAY command shows the console group definitions in effect for the sysplex.

```
D CNGRP[, {GROUP|G} [= (name[,name]...)] ][,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows console status information.

DISPLAY Command

```

D {CONSOLES},{{ACTIVE|A }[,CA={name          }][,ROUT={NONE|ALL|rr }]}
{C          } {{SS          } {name[,name]...}} {(rr-ss) } }
              {{NACTIVE|N}[MSTR]           } {(rr[,ss]...)} }
              {                               [,SYS=system name] }

              {KEY[=key]                      }

              {{BACKLOG|B}                    }

              {{MASTER|M}[,SYS=system name]   }

              {{MCONLY}                       }

              {*                               }

              {{LIST|L}                       }

              {{HARDCOPY|HC}[,SYS=system name]}

              {CN={xx          }[,ROUT={NONE|ALL|rr }]}
              {  {(xx[,yy]...)} {(rr-ss) } }
              {  {(xxx-yyy) } {(rr[,ss]...)} }
              {                               [,SYS=system name] }

              {HCONLY                          }

              {U={{[/]devnum1[,[/]devnum2]...}} }
              {  {{[/]lowdevnum-[/]highdevnum}} }
              {  {[/]devnum } }
              {MSTR[,SYS=system name] }
              {SMCS                             }

              [,L={a|cc|cca|name|name-a}]

```

Example: The following DISPLAY command shows a list of major names or resource information for the specified resource(s).

```

D DLF[,RES={{(qname|*)[,rname|,*]}}][,HEX][,L={a|cc|cca|name|name-a}]

```

Example: The following DISPLAY command shows the domain description table or a specific table entry (*domainnum* from 2 to 129).

```

D DMN[=domainnum][,L={a|cc|cca|name|name-a}]

```

Note: The above command is not valid on systems operating in workload management goal mode.

Example: The following DISPLAY command shows the dump options or dump data set status.

```

D {DUMP},{{STATUS|ST|S}                                     }
  {D      }
    {{OPTIONS|O}                                           }
      {{TITLE|T      }},{AUTODSN={aaa|ALL}                 }
      {{ERRDATA|ER|E}{                                     }
        {,DSN={ALL|(ALL)                                   }
          {nn|(nn[,nn]...)                                }
          {nn-nn|(nn-nn[,nn-nn]...)                       }
          {(nn[,nn]...[,nn-nn[,nn-nn]...)                 }
        }
      {,DUMPID={xxx|(yyy[,zzz]...)                         }
        {aaa-bbb|(aaa-bbb[,ccc-ddd]...)                   }
        {(yyy[,zzz]...[,aaa-bbb[,ccc-ddd]...)             }
      }

[,L={a|cc|cca|name|name-a}]

```

Example: The following DISPLAY command shows information about extended MCS (EMCS) consoles.

```

D EMCS,{SUMMARY|S}[,L={a|cc|cca|name|name-a}]
  {INFO|I}
  {FULL|F}

  {STATUS=A|N|L|B[{}]|ERR }
  {ST      }

  {CN=consname|*      }
  {SYS=sysname|*      }
  {KEY=keyname|*      }
  {AUTH={ANY}
    {MASTER}
    {SYS}
    {IO}
    {CONS}
    {ALL}
    {INFO}
    {SYSONLY}rbrc;
    {IOONLY}
    {CONSONLY}
    {ALLOONLY}
    {INFOONLY}
  }

  {ATTR={ANY}
    {YES}
    {ROUT}
    {UD(YES|NO)}
    {HC}
    {AUTO(YES|NO)}
    {MN}
    {NONE}
  }

  {DOM={ANY}
    {NORMAL}
    {ALL}
    {NONE}
    {YES}
  }

```

DISPLAY Command

Example: The following DISPLAY command shows current external timer reference (ETR) synchronization.

```
D ETR[,DATA] [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows global resource serialization (GRS) information.

```
D GRS{[,SYSTEM|,LINK|,ALL|,A|,DELAY|,D|,SUSPEND|S]}
      [,CONTENTION|,C]
      [,RES=(qname|*[,rname|,*])]
      [,HEX]
      [,DEV=[/]devnum] [, {SUSPEND|S}]
      [,RNL={CONVERSION|CON|C}]
          {ALL|A }
          {EXCLUSION|EXCL|E}
          {INCLUSION|INCL|I}

      {,CONTENTION|C[,ENQ] [,LATCH[, {JOBNAME|JOB}=jobname]] [,HEX]}

      {,LATCH, {JOBNAME|JOB}=jobname] [,CONTENTION|,C]} [,HEX]
          {CONTENTION|C}}

      {,ANALYZE|,ANALYSE|,AN, }
          {BLOCKER|BLOCK }
          {WAITER|WAIT[,SYSTEM|,SYS=sysname|*]} }
              [,ASID|,AS=asid]
              [,JOBNAME|,JOB=jobname]
              [,XJOBNAME|,XJOB=(jobname1,jobname2,...,jobname25)]
              [,XQNAME|XQN=(qname1,qname2,...,qname25)]

          {DEPENDENCY|DEPEND|DEP[,SYSTEM|,SYS=sysname|*]} }
              [,ASID|,AS=asid] [TCB=tcbaddr]
              [,JOBNAME|JOB=jobname]
              [,XJOBNAME|,XJOB=(jobname1,jobname2,...,jobname25)]
              [,XQNAME|,XQN=(qname1,qname2,...,qname25)]

              [,RES=(qname,rname)]
                  {SCOPE|SCO=SYSTEM[S]|SYS[S]}
                  {SYSTEM|SYS=sysname|*}

          [,COUNT|,CNT=nn] [,DETAIL|,DET]

      [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows IOS-related configuration information.

```
D IOS,CONFIG[(EDT)|(HSA)|(ALL)] [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows dynamic channel path management information.

```
D IOS,DCM
```


Example: The following DISPLAY command shows XCF group information.

```
D IOS,GROUP
```

Example: The following DISPLAY command shows missing interruption handler (MIH) time intervals.

```
D IOS,MIH[,TIME={ALL|option} ]
[ ]
[, {DEV }={([/]devnum[,[/]devnum1]...)} ]
[ {DEVX } {([/]devnum-[/]devnum1[,[/]devnum2-[/]devnum3]...)} ]
[ {TDEV } ]
[ {TDEVX} ]

[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows information about the devices the IOACTION STOP command affects.

```
D IOS,STOP[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows information about the system's initial program loading (IPL).

```
D IPLINFO [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows system activity information.

```
D {JOBS|J|A|TS}[, {LIST|L}, [USERID=userid]]
[, {ALL|A} ]
[, {jobname[.identifier] | (jobname)} ]

[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows how to request library lookaside information.

```
D LLA
```

Example: The following DISPLAY command shows the status of the system logger, individual log streams, or log streams from a sysplex view.

DISPLAY Command

```

D  LOGGER[ ,STATUS | ST ]
      [,Connection[,LSName=logstreamname[ [,Jobname=mvsjobname] [,SUMM ] ] ]
      [,Detail]
      [,Jobname=mvsjobname[ [,LSNAME|,LSN=logstreamname] [,SUMM ] ]
      [,Detail]
      [,SYSPLEX[,LSName=logstreamname]
      [,DASDONLY
      [,Logstream[,LSName=logstreamname] [,STRNAME|,STRN=structurename]
      [,DASDONLY
      [,STRUCTURE|,STR[,STRNAME|STRN=structurename] ]

```

Example: The following DISPLAY command shows the logrec error and environmental record recording medium.

```

D  LOGREC[ ,{CURRENT|CURR} | {DATASET|DSN} | {ALL|A} ]
      [,L={a|cc|cca|name|name-a} ]

```

Example: The following DISPLAY command shows system configuration information.

```

D  M[ =CHP[ (xx) | (xx-xx) | (list) ]
      | =CONFIG[ (xx) ]
      | = {CPUID|CPU} [ (x) | (list) ]
      | = {DEVICE|DEV} [ ( [/] devnum ) | ( [/] lowdevnum- [/] highdevnum ) | (list) ]
      | = {DEVICE|DEV} ( [/] devnum, (chp) )
      | = {DEVICE|DEV} ( ( [/] devnum ), (chp) )
      | = ESTOR[ (ddddM-ddddM) | (list) | (E [=id]) ]
      | = HIGH
      | = HSA
      | = SIDE[ (id) ]
      | = {STORAGE|STOR} [ (ddddM-ddddM) | (list) | (E [=id]) ]
      | = SWITCH( sssss [ ,pp[-pp] [ ,pp[-pp] ] ... )
      | = (parm[,parm] ... )
      [,L={a|cc|cca|name|name-a} ]

```

Example: The following DISPLAY command specifies the status of the MVS message service and a list of languages available.

```

D  MMS[ ,L={a|cc|cca|name|name-a} ]

```

Example: The following DISPLAY command shows information about message processing and presentation.

DISPLAY Command

```
D MPF[, {MSG|M} ] [,L={a|cc|cca|name|name-a}]
    |, {COLOR|C}
    |, CMD
```

Example: The following DISPLAY command shows information about z/OS UNIX.

```
D OMVS[{,SUMMARY|S} ]
    |, {ASID|A}=ALL
    |, {ASID|A}=asid
    |, U=userid
    |, {PID}=processid[,BRL]
    |, {FILE|F[,CAPS|C]}
    |, {VSERVER|V}
    |, {PFS|P}
    |, {CINET|CI}=A11|TPname
    |, {OPTIONS|O}
    |, {LIMITS|L[,PID=ProcessId] [,RESET]}

[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows sysplex-wide operator information.

```
D {OPDATA|O} [,PREFIX] [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows the parmlib data sets and volume serial numbers that are defined in LOADxx and the MASTER JCL (when there are no LOADxx parmlib statements).

```
D PARMLIB [,ERRORS|E] [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows PFK definitions and tables.

```
D PFK[,CN=cc ] [,L={a|cc|cca|name|name-a}]
    |, {TABLE|T} [=nnnnnnnn]
```

Example: The following DISPLAY command shows information about registered products or the product enablement policy.

```
D PROD, {REGISTERED|REG|STATE|STATUS}
    [,OWNER(o)] [,NAME(n)] [,FEATURENAME(fn)] [,ID(id)] [,ALL]
```

DISPLAY Command

Example: The following DISPLAY command shows entries in the list of APF-authorized libraries.

```
D PROG,APF[,ALL ][,L={a|cc|cca|name|name-a}]
    ,DSNAME=libname
    ,ENTRY=xxx
    ,ENTRY=(xxx-yyy)
```

Example: The following DISPLAY command shows the dynamic exits.

```
D PROG,EXIT,{ {EXITNAME|EX|EN}=exitname }[,DIAG]
    { {EXITNAME|EX|EN}=exitname* }
    { {MODNAME|MOD}=modname }
    { [ALL] [,IMPLICIT|,IMP] }

[,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows information about LNKLST sets for the LNKLST concatenation and associated jobs.

```
D PROG,LNKLST[,NAME=[lnk1stname|CURRENT] ]
    [,NAMES ]
    [,USERS,[CURRENT|NOTCURRENT|NAME=lnk1stname]]
    [,ASID=asid ]
    [,JOBNAME=jobname ]
```

Example: The following DISPLAY command shows the entry point, load point, and size information about modules in the LPA, and shows the minimum amount of CSA and ECSA that must remain after dynamically adding a module to the LPA.

```
D PROG,LPA{,MODNAME=modname}[,L={a|cc|cca|name|name-a}]
    {,CSAMIN }
```

Example: The following DISPLAY command shows information about outstanding action messages (descriptor codes 1, 2, 3, and 11), WTORS, devices awaiting mount requests to be fulfilled, and units requiring intervention.

DISPLAY Command

```

D R[,U]
  ,KEY[,SYS=sysname] [,CN=(ALL)]
  [,I] [,msgformat] [,MSG=msgid] [,SYS=sysname] [,KEY=keyname]
  [,] [,JOB=jobname]
  ,E [,CN={xx|name|(ALL)}] [,ROUT={ALL|(rrr[,sss]...) }
  ,CE {(rrr-sss[,rrr-sss]...)}
  ,R
  ,M
  ,{LIST|L}
  ,{ALL|A}
  ,
  (See Note)
  [,L={a|cc|cca|name|name-a}]

```

Note: If you supply all commas between DISPLAY R and the operands that have equal signs, you get default values. However, you should supply only one comma before the L operand, even if you omit the preceding operands.
Example: DISPLAY R,I,L=2B

Example: The following DISPLAY command shows the current status of the run-time library services (RTLS) environment. It lists information about the physical and logical libraries in use, the users of the logical libraries, and the cache use for a given library or for all libraries.

```

D RTLS[,NAMES[,LIBRARY=1name[,VERSION=ver] [,CURRENT|,SEQNUM=num|,ALL] ]
  [,PHYSICAL,LIBRARY=pname[,CURRENT|,SEQNUM=num|,ALL] [,MODULE=mod|,LOGICAL]]
  [,LOGICAL{,LIBRARY=1name[,VERSION=ver] [,CURRENT|,SEQNUM=num|,ALL] } ]
  { [,MODULE=mod|,USERS] }
  { ,JOBNAME=jobname }
  { ,ASID=asid }
  [,L={a|cc|cca|name|name-a}]

```

Example: The following DISPLAY command shows detailed information about the SLIP trap identified by identifier **xxxx**. If **xxxx** is not specified, the system displays all traps and tells whether each is enabled.

```

D SLIP[=xxxx] [,L={a|cc|cca|name|name-a}]

```

Example: The following DISPLAY command shows the SMF data set names and their status (**S**) or the current SMF options (**O**).

```

D SMF[,S|,O] [,L={a|cc|cca|name|name-a}]

```

Example: The following DISPLAY command shows storage management subsystem (SMS) information.

DISPLAY Command

```

D SMS[, {ACTIVE|A} ]
    [, CACHE ]
    [, CFCACHE(structurename|*) ]
    [, CFLS ]
    [, CFVOL(void) ]
    [, {DRIVE|DRI} (name|ALL) [, STATUS ] ]
    [ |,DETAIL ] ]
    [, {LIBRARY|LIB} (name|ALL) [, STATUS[, LISTDRI]] ]
    [ |,LISTDRI ] ]
    [ |,DETAIL ] ]
    [, MONDS(specmask|*) ]
    [, OAM ]
    [, OPTIONS ]
    [, OSMC[, TASK(name)] ]
    [, SHCDS ]
    [, SMSVSAM[, ALL] ]
    [, SMSVSAM, QUIESCE ]
    [, {STORGRP|SG} {(storgrp|ALL)} [, LISTVOL ] ]
    [ |,DETAIL ] ]
    [, {TRACE|T} ]
    [, {VOLUME|VOL} (volume) ]
    [, L={a|cc|cca|name|name-a}]

```

Example: The following DISPLAY command shows information about all subsystems defined to MVS.

```

D SSI[, {LIST|L} | {ALL|A}] [, {DYNAMIC|DYN|D}={YES|Y} | {NO|N}]
    [, {FUNC|F}=func] list ]
    [, {STATUS|STAT|ST}={ACTIVE|ACT} | {INACTIVE|INACT|I} ]
    [, {SUBSYS|SUB}=subsysname ]
    [, L={a|cc|cca|name|name-a}]

```

Example: The following DISPLAY command shows static system symbols and their associated substitution text.

```

D SYMBOLS[, L={a|cc|cca|name|name-a}]

```

DISPLAY Command

Example: The following DISPLAY command shows the local time of day and date and Greenwich Mean Time (GMT).

```
D T
```

Example: The following DISPLAY command shows component trace status.

```
D TRACE[,COMP=cname[,SUB=(subname)][,N=nnn] [,SUBLEVEL] ]
      [,COMP={(cname[,cname]...) | ALL} ]
      [,WTR={(name[,name],...) | ALL} ]
      [,TT ]
      [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows device status and allocation.

```
D {U[,devicetype] [,ONLINE ] [[, [/] devnum] [,nnnn] ] }
   [,OFFLINE]
   [,ALLOC ] }
   [,AUTOSWITCH|AS] [[[, [/] devnum] [,nnnn] [SYS=sysname]]] (See Note)}
{U,IPLVOL }
{U,VOL=volser }
[,L={a|cc|cca|name|name-a}] }
```

Note: Supply all commas between DISPLAY U and a specified operand.

Example: The following DISPLAY command shows the workload management service policy, service definition, mode, application environments, resources, and scheduling environments.

```
D WLM[,SYSTEM=sysname|,SYSTEMS]
     [,APPLENV=applenvname|*]
     [,SCHENV=schenvname[,SYSTEM=sysname|,SYSTEMS]]
     [,RESOURCE=resourcename[,SYSTEM=sysname|,SYSTEMS]]
     [,L={a|cc|cca|name|name-a}]
```

Example: The following DISPLAY command shows a summary of the current sysplex.

DISPLAY Command

```

D XCF[, {PATHIN|PI} ]
    [, {DEVICE|DEV}={ ([/] indevnum[, [/] indevnum...)] |ALL} ]
    [, {STRNAME|STRNM}={ (strname[, strname]...) |ALL} ]
    [, {SYSNAME|SYSNM}=(sysname[, sysname]...) ]
    [, {STATUS|STAT}=( [STARTING] [, RESTARTING] [, WORKING]
        [, STOPPING] [, STOPFAILED] [, INOPERATIVE]
        [, LINKING] [, QUIESCING) ) ]

    [, {PATHOUT|PO} ]
    [, {DEVICE|DEV}={ ([/] outdevnum[, [/] outdevnum...)] |ALL} ]
    [, {STRNAME|STRNM}={ (strname[, strname]...) |ALL} ]
    [, CLASS={ (classname[, classname]...) |ALL} ]
    [, {SYSNAME|SYSNM}=(sysname[, sysname]...) ]
    [, {STATUS|STAT}=( [STARTING] [, RESTARTING] [, WORKING]
        [, STOPPING] [, STOPFAILED] [, INOPERATIVE]
        [, LINKING] [, QUIESCING) )
        [, REBUILDING] [, QUIESCED) ) ]

    [, {LOCALMSG|LM} [, CLASS={ (classname[, classname]...) |ALL} ] ]

    [, {GROUP|GRP} [, groupname[, membername|ALL] ] ]

    [, {SYSPLEX|S} [, systemname|ALL] ]

    [, {COUPLE|CPL} [, TYPE={ (name[, name]...) |ALL} ] ]

    [, {CLASSDEF|CD} ]
    [, CLASS={ (classname| (classname[, classname]...) |ALL} )
        [, {GROUP|G}=groupname ]

    [, {STRUCTURE|STR} ]
    [, {STRNAME|STRNM}={ (strname[, strname]...) |ALL} ]
    [, {CONNAME|CONNM}={ (conname[, conname]...) |ALL} ]
    [, {STATUS|STAT}=( [ALLOCATED] [, NOTALLOCATED]
        [, POLICYCHANGE] [, DEALLOCPENDING]
        [, LARGERCFRMDS] [, REBUILD] [, STRDUMP]
        [, ALTER] [, FPCONN] [, NOCONN) ) ]

    [, {CF} [, {CFNAME|CFNM}={ (cfname[, cfname]...) |ALL} ] ]

    [, {POLICY|POL} [, TYPE={ (name[, name]...) |ALL} ] ]

    [, {PRSMPOLICY|PRSMPOL} ]

    [, {ARMSTATUS|ARMS} ]
    [, {RESTARTGRP|RG}=rgname ]
    [, {ELEMENT|EL}=elname| {JOBNAME|JOB}=jobname ]
    [, INITSYS=initsys ]
    [, CURRSYS=currsys ]
    [, STATE=( [STARTING|START] [, {AVAILABLE|AVAIL} [, FAILED]
        [, {RESTARTING|RESTART} [, {RECOVERING|RECOVER} ) ) ]
    [, DETAIL ]

    [, L={ a|cc|cca|name|name-a } ]

```


DUMP Command

Purpose: Use the DUMP command to request a system dump of virtual storage (SVC dump). The system responds by prompting you in message IEE094D for the dump options. Specify these options by using the REPLY command. The title of the dump, which you specify by enclosing 1-100 characters in parentheses or single or double quotes, becomes the first record in the dump data set. COMM and TITLE are synonyms.

```
DUMP
DUMP {COMM={ (title) } [,PARMLIB=xx]          }
      { 'title' }   [,PARMLIB=(xx[,xx]...)]
      { "title" }  [SYMDEF=(symdef[,symdef]...)]
      {TITLE={ (title) }                       }
      { 'title' }
      { "title" }
```

DUMPDS Command (DD)

Purpose: Use the DUMPDS command to:

- Change the system's list of dump data sets and resources
- Clear full SYS1.DUMP data sets and make them available for dumps
- Set up or alter the configuration of automatic dump data set allocation

Example: The following DUMPDS command adds the named direct access data sets to the list of SYS1.DUMP data sets.

```
DD ADD, {DSN={nn                               } }
          { (nn[,nn]...)                       }
          { nn-nn                               }
          { (nn-nn[,nn-nn]...)                 }
          { (nn[,nn]...,nn-nn[,nn-nn]...) }
          { ALL                                 }
          { (ALL)                              }

          {SMS={class                           } }
            { (class[,class]...) }

          {VOL={volser                           } }
            { (volser[,volser]...) }
```

Where **class** represents:

```
{storclas
 { ([DATA]D=[dataclas] [,MGMT|M=[mgmtclas] [,STOR|S=[storclas]]) }
```

Example: The following DUMPDS command specifies whether or not to allocate dump data sets automatically when requesting a dump.

```
DD ALLOC={ACTIVE|INACTIVE}
```

DUMPDS Command

Example: The following DUMPDS command empties the specified data set and marks it available to receive a dump.

```
DD CLEAR,DSN={nn                }
              {(nn[,nn]...)     }
              {nn-nn            }
              {(nn-nn[,nn-nn]...) }
              {(nn[,nn]...,nn-nn[,nn-nn]...)}
              {ALL               }
              {(ALL)             }
```

Example: The following DUMPDS command removes from the system's list of dump data set resources specific SYS1.DUMP data sets, SMS classes, or DASD volumes.

```
DD DEL,{DSN={nn                } }
        {(nn[,nn]...)          }
        {nn-nn                 }
        {(nn-nn[,nn-nn]...)    }
        {(nn[,nn]...,nn-nn[,nn-nn]...)}
        {ALL                   }
        {(ALL)                 }

        {SMS={class            } }
          {(class[,class]...)   }
          {ALL                   }
          {(ALL)                 }

        {VOL={volser           } }
          {(volser[,volser]...) }
          {ALL                   }
          {(ALL)                 }
```

Where **class** represents:

```
{storclas
{([DATA|D=[dataclas]] [,MGMT|M=[mgmtclas]] [,STOR|S=[storclas]])}
```

Example: The following DUMPDS command establishes a name pattern for automatically allocated dump data sets.

```
DD NAME=name-pattern
```

FORCE Command

Purpose: Use the FORCE command to terminate a named batch job, started task, or APPC/MVS transaction program. **CAUTION:** Use this command as a last resort when the CANCEL command fails to perform its function after you have issued it several times.

FORCE
<pre>FORCE {jobname } [,ARM] [,A=asid] [,ARMRESTART] {U=userid } {[jobname.]identifier}</pre>

WARNING: BE SURE YOU ARE AWARE OF ALL THE CONSEQUENCES OF ISSUING A FORCE COMMAND.

HALT Command (Z)

Purpose: Use the HALT command to record statistics before stopping the operating system. Store the internal I/O device error counts in the logrec data set, empty the SMF buffers on to the active SMF data set in SYS1.MANx, close the system log, and put it on the print queue.

HALT or Z
Z EOD

IOACTION Command (IO)

Purpose: Use the IOACTION command to stop and resume I/O activity to direct access storage devices (DASDs) without varying the DASD offline, when the DASD is shared between systems AND is in recovery by the input/output system (IOS). **CAUTION:** Use this command only in response to the IOS recovery messages IOS427A and IOS062E and wait state X'062'.

IOACTION or IO
<pre>IO {STOP,DEV=([/]devnum[, [/]devnum]...) } {STOP,DEV=([/]lowdevnum-[/]highdevnum[, [/]lowdevnum-[/]highdevnum]...) } {RESUME,DEV=([/]devnum[, [/]devnum]...) } ALL {RESUME,DEV=([/]lowdevnum-[/]highdevnum[, [/]lowdevnum-[/]highdevnum]...) }</pre>
<p>Note: You can enter individual device numbers and ranges on the same command. For example:</p> <pre>IO RESUME,DEV=(/2233,/990-/1012,160)</pre>

LIBRARY Command (LI)

Purpose: Use the LIBRARY command to perform any of several tasks associated with tape drives and tape volumes.

For a detailed discussion of the LIBRARY command, including the parameters, syntax, and the tasks that command can perform, refer to *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* and *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

LOG Command

LOG Command (L)

Purpose: Use the LOG command to make an entry into the system log (up to 122 characters), the OPERLOG, or both. The entry goes to the master console if the log is temporarily inactive.

LOG or L
L 'text'

LOGOFF Command

Purpose: Use the LOGOFF command to log off from an MCS console. This command will end your terminal session.

LOGOFF
LOGOFF

Notes:

1. You must issue LOGOFF when you leave your console and your installation requires operators to log on before issuing commands or performing secured functions.
2. When your installation requires a LOGON command, LOGOFF leaves the console in a secure state. The system does not accept commands from this console until another LOGON command is completed. The console is in roll mode (MODE=R).
3. When LOGON is automatic at your installation, the system issues another MCS LOGON command to that MCS console.

LOGON Command

Purpose: Use the LOGON command to identify yourself to the system when your installation requires operators to log on before issuing commands.

LOGON
IEE187I ENTER LOGON PARAMETERS
LOGON {userid} PASSWORD {password}
GROUP [racfgroup] SECLABEL [label]

MODE Command

Purpose: Use the MODE command to control the actions of recovery management when certain types of machine check interruptions occur.

Example: The following MODE command controls the recording of hard machine check interruptions.

```

MODE {PD} [, INTERVAL={nnnn}] [, RECORD[=nnn ] [, CPU={x } ]
    {SD}           {300 }           | =ALL      {ALL}
    {IV}           | =25
    {TC}           | =16
    {PT}           | =5
    {CC}
    {VS}
    {PS}
    {AD}
    {SL}
    {SC}

```

Example: The following MODE command controls the recording and reporting of system recovery and degradation machine check interruptions.

```

MODE {SR} [, QUIET ] [, CPU={x|ALL} ]
    {DG} [, RECORD[=nnn ] [, REPORT=nnn] ]
        | =ALL
        | =50
        | =1

```

Example: The following MODE command displays the event counters and recording monitoring status associated with each type of machine check interruption.

```

MODE [STATUS]

```

MODIFY Command (F)

Purpose: Use the MODIFY command to pass information to a job or started task. You can communicate with a currently running program only if it was designed to recognize input from the MODIFY command.

Example: The following MODIFY command changes the job parameters as the programmer specifies.

```

F [jobname.]identifier,parameters

```

Example: The following MODIFY command starts TSO/TCAM time-sharing once TCAM is active.

```

F [jobname.]identifier,TS=START[,member]

```

Example: The following MODIFY command stops TSO/TCAM time-sharing.

MODIFY Command

```
F [jobname.]identifier,TS=STOP
```

Example: The following MODIFY command modifies TSO/VTAM time-sharing.

```
F [jobname.]identifier,{USERMAX=nnnnn}
                        {USER={SIC }}
                        {      {FSTOP} }
```

Example: The following MODIFY command controls z/OS UNIX System Services, terminates an z/OS UNIX process or thread, shuts down the z/OS UNIX initiators, or requests a SYSMDUMP for a process.

```
F BPX0INIT,{APPL=appl_data}
           {DUMP=pid}
           {FILESYS={DISPLAY[, FILESYSTEM=filesystemname] [,OVERVERRIDE]}
           |,ALL
           |,EXCEPTION
           |,GLOBAL
           {DUMP                                }
           {FIX                                  }
           {REINIT                              }
           {RESYNC                              }
           {UNMOUNT,FILESYSTEM=filesystemname }
           {UNMOUNTALL                          }
           {FORCE=pid[.tid]}
           {RESTART=FORKS}
           {SHUTDOWN={FILESYS | FORKINIT | FORKS}}
           {TERM=pid[.tid]}
```

Example: The following MODIFY command communicates with the catalog address space. Use this command only at the direction of the system programmer.

```

F CATALOG,{ABEND{(id)          }          }
           { (yyyyyyyy)        }          }
           { {ALLOCATE}        }          }
           { {ANALYSIS}        }          }
           { {MODIFY}          }          }
           {ALIASLEVEL(n)      }          }
           {ALLOCATE(catname)[,{NOISC | NOVLF}] }
           {ALLOCATED[(vvvvvv)] }
           {CATMAX(nnnn)       }          }
           {CLOSE(catname)     }          }
           {{DUMPON | DUMPOFF} }          }
           {ECSHR({(ADD,catname) }) }
           | (REMOVE,catname)
           | (STATUS,catname)
           {ECSHR(AUTOADD)     }          }
           {ECSHR({CONNECT   }) }
           | DISCONNECT
           | STATUS
           {END(id)[,{REDRIVE | NOREDRIVE}] }
           {ENTRY[(cname) | (mmmmmm)] }
           {{ISC | NOISC}(catname) }
           {LIST[(id) | (yyyyyyyy)] }
           {LISTJ[jobname]     }          }
           {NOTIFYEXTENT(xxx)   }          }
           {OPEN[(vvvvvv)]     }          }
           {REPORT              }          }
           {REPORT,CACHE[(catname)] }
           {REPORT,DUMP        }          }
           {REPORT,PERFORMANCE[(RESET)] }
           {RESTART            }          }
           {{ROTATE | NOROTATE} }
           {{SYS%ON | SYS%OFF} }
           {TASKMAX(nnn)       }          }
           {UNALLOCATE[catname] }
           {VCLOSE(vvvvvv)     }          }
           {{VLF | NOVLF}(catname) }
           {{VUNALLOCATE | NOVUNALLOCATE} }
           {{WARNING | NOWARNING} }
    
```

Example: The following MODIFY command changes the processing mode for the data lookaside facility (DLF).

```

F DLF,MODE={DRAIN|D }
           {QUIESCE|Q}
           {NORMAL|N }
    
```

Example: The following MODIFY command causes the data lookaside facility (DLF) to use the specified COFDLFxx member of SYS1.PARMLIB.

```

F DLF,NN=xx
    
```

Example: The following MODIFY command displays the limits from the COFDLFxx parmlib member currently in effect.

MODIFY Command

```
F DLF[, {STATUS|ST|S}]
      |,SM
      |,SB
```

Example: The following MODIFY commands builds and replaces library lookaside (LLA) directories.

```
F LLA, {REFRESH }
      {UPDATE=xx}
```

Example: The following MODIFY command operates with the network file system server.

```
F [MVS NFS.|jobname.]identifier,
   {FREEZE={ON|OFF}           }
   {LIST={MOUNTS|DSNAMES}     }
   {RELEASE=datasetname[(member)] }
   {STATUS                     }
   {STOP                       }
   {UNMOUNT=name              }
```

Example: The following MODIFY command diagnoses problems for the network file system server.

```
F [MVS NFS.|jobname.]identifier, LOG={ERROR|WARN|INFO|MEMSTATS}
```

Example: The following MODIFY command displays information about the object access method (OAM) or requests that OAM perform a specified service: object management, space management, or recovery functions.

MODIFY Command

```
F [OAM.|jobname.]identifier,
  {{LABEL|L}[,9247          ]
    |,3995-133
    |,3995WORM
    |,3995REWR
  }
  {{START|S},{OSMC
    {STORGRP,stogrprname}
    {LIBMGT,libraryname}
    {DASDSM,stogrprname}
    {RECOVERY,volser}
    {OBJRECV,collectionname,objectname}
    {MOVEVOL,volser}
  }}
  {{STOP|P},{OAM
    {OSMC}
    {STORGRP,stogrprname}
    {MOVEVOL,volser}
  }}
  {{DISPLAY|D},{GROUP[,stogrprname]}
    {VOL,volser}
  }
```

Example: The following MODIFY command switches workload management modes and changes resource settings.

```
F WLM,[MODE={GOAL|COMPAT}]
  ,[RESOURCE=resource, {ON|OFF|RESET}]
```

Example: The following MODIFY command specifies the criteria that an external writer is to use in selecting data sets for processing.

```
F [XWTR.|jobname.]identifier,
  {{CLASS|C}=[classes]
  }
  {{DEST|D}=[LOCAL
    |remote-workstation-name
  ]}
  {{FORMS|F}=[forms-name]
  }
  {{JOBID|J}=[JOBnnnn ]
    |STCnnnn
    |TSUnnnn
  }
  {{WRITER|W}=[STDWTR
    |user-writer-name
  ]
  }
```

Example: The following MODIFY command causes an external writer to pause.

MODIFY Command

```
F [XWTR.|jobname.]identifier,{PAUSE|P}={FORMS }
                                     {DATASET}
```

MONITOR Command (MN)

Purpose: Use the MONITOR command to display jobnames, data set status, and time-sharing user sessions continuously, and to add certain information to mount and demount messages.

MONITOR or MN

```
MN {JOBNAMES[,T][,L={a|cc|cca|name|name-a}] }
   {DSNAME }
   {SPACE }
   {STATUS[,L={a|cc|cca|name|name-a}] }
   {SESS[,T][,L={a|cc|cca|name|name-a}] }
```

MOUNT Command (M)

Purpose: Use the MOUNT command to allow allocation of an I/O device to all job steps that require a particular volume, without intervening demountings and remountings of the volume.

MOUNT or M

```
M {[/]devnum },VOL=({NL},serial)[,USE={STORAGE}]
  {devicetype } {SL} {PUBLIC }
                 {AL} {PRIVATE}
```

MSGRT Command (MR)

Purpose: Use the MSGRT command to establish or change message routing instructions for displays from the DISPLAY, TRACK, or CONFIG commands to a specified message area, console, or both. MSGRT also controls the action of the CONTROL, MONITOR, STOPTR, and STOPMN commands.

MSGRT or MR

```
MR { [D=(operand[,operand]...)] [,L={a } ] }
    |TR=A {cc }
    |K {cca }
    |CF {name }
    |MN {name-a }
    {REF }
    {NONE }
    {CONTn }
```

PAGEADD Command (PA)

Purpose: Use the PAGEADD command to add auxiliary storage space (local page data sets) to the system.

CAUTION: Use this command only at the direction of your system programmer.

PAGEADD or PA
<pre>PA { [PAGE=] } { dsname[, dsname] ... } {NONVIO=}</pre>

PAGEDEL Command (PD)

Purpose: Use the PAGEDEL command to delete, replace, or drain (quiesce) local page data sets. **WARNING:** Use this command only at the direction of your system programmer. Misuse can seriously impact system performance by causing a shortage of auxiliary storage.

PAGEDEL or PD
<pre>PD {DELETE, PAGE={ dsname[, dsname] ... } {REPLACE, PAGE={ (dsname, rdsname) [, (dsname, rdsname)] ... } {DRAIN, PAGE={ dsname[, dsname] ... }</pre>

QUIESCE Command

Purpose: Use the QUIESCE command to put the system in a manual state without affecting job step timing.

QUIESCE
<pre>QUIESCE</pre>

REPLY Command (R)

Purpose: Use the REPLY command to respond to system requests for information. The verb (REPLY, or R) is not required when you respond to a request. The general syntax for the REPLY command is:

<pre>[R] id[,] ['text' text]</pre>

Example: The following REPLY command would be your response to system requests during recovery processing.

REPLY Command

```
R [00|0] [,] ['text' | text]
```

Example: The following REPLY command sets the time of day clock. If you specify GMT, the system assumes Greenwich Mean Time. Otherwise, it assumes local time.

```
R 00, ' [DATE=yyyy.ddd] [,CLOCK=hh.mm.ss] [,GMT] '
```

Example: The following REPLY command would be your response to message ITT006A, which the system issues after you issue a TRACE CT (component trace) command, to prompt you for the options you want to specify.

```
R id[,ASID=(nnnn[,nnnn]...)]  
    [,JOBNAME=(name[,name]...)]  
    [,OPTIONS=(name[,name]...)]  
    [,WTR={membername|DISCONNECT}]  
    [,CONT|,END ]
```

Note: When you specify CONT or END, it must be the last parameter on the input line.

Example: The following REPLY command would be your response to message IEE094D, which the system issues after you issue a DUMP command, to prompt you for the dump options you want to specify.

```
R id,U  
  
or  
  
R id[,ASID=(n[,n]...)] [,JOBNAME=(name[,name]...)] [,TSONAME=(name[,name]...)]  
    [,DSPNAME=(dspname-entry[,dspname-entry]...)]  
    [, {PROBDESC|PROB|PD}=key-spec] [,REMOTE=(request[,request]...)]  
    [,SDATA[(option[,option]...)] [,STOR=(beg,end[,beg,end]...)]  
    [,STRLIST=(s-option[,s-option]...)]  
    [,CONT|,END]
```

Notes:

1. When you specify CONT or END, it must be the last parameter on the input line.
2. The CONT keyword does not work within a SYSP= list.
3. When you specify U, it must be the first parameter following the identification number.

Where **request** represents:

```
{GRPLIST={group(member) } }  
        {(group(member[,member]...)[,group(member[,member]...)]...)}  
  
{SYSLIST={sysinfo| (sysinfo[,sysinfo]...)} }  
        [,DSPNAME|,DSPNAME=(dspname-entry[,dspname-entry]...)]  
        [,SDATA|,SDATA=(option[,option]...)]  
        [,STOR|,STOR=(beg,end[,beg,end]...)]
```

Where **s-option** represents:

```

STRNAME=strname
[,CONNAME=conname ]
[,ACCESSTIME={ENFORCE|NOLIMIT|NOLIM} ]
[,LOCKENTRIES ]
[,USERCNTLS ]
[,EVENTQS ]
[, (EMCONTROLS={ALL|(list)}) ]
[, ({COCLASS|STGCLASS|LISTNUM}={ALL|(list)}
  {[,ADJUNCT={CAPTURE|DIRECTIO}] [,ENTRYDATA={UNSERIALIZE|SERIALIZE}}]
  {[,SUMMARY] }) ]

```

Example: The following REPLY command would be your response to message IEE357A, which the system issues to allow you either to specify that you do not want to change the SMF values (U), or to specify the options (separated by commas) your system programmer provides.

```

R id,{U
  {options}

```

Example: The following REPLY command would be your response to message IEA101A, which the system issues during system initialization, to prompt you to specify system parameters.

```

R [0|00],{U
  {
    {'parm=',' } [,CONT] }
    {'parm=,parm,' } }
    {'parm=value' } }
    {'parm=(value[,value]...[,L])' } }
    {'parm=(value[,value]...[,L]),parm=value' } }
  }

```

RESET Command (E)

Purpose: Use the RESET command to force a hung MCS console offline or to change certain aspects of work currently executing.

Example: The following RESET command forces a hung MCS console offline.

```

E [CN(consname)]

```

RESET Command

Example: The following RESET command can change the performance group or service class of an executing job, time-sharing user, or started task. Also, in goal mode only, it can quiesce a problem job or address space or resume a quiesced job or address space. **CAUTION:** Use this command only at the direction of the system programmer.

```
E jobname[,A=asid],{PERFORM=nnn      }
                    {SRVCLASS=classname}
                    {QUIESCE|Q      }
                    {RESUME          }
```

ROUTE Command (RO)

Purpose: Use the ROUTE command to direct a command to one or more systems in a sysplex for processing.

```
ROUTE or RO

RO {sysname,text                                     }

  {[T=nnn,]*ALL                                     }[,L={a    }] }
    {sysgrpname                                     }   {cc    }
    {*OTHER                                         }   {cca   }
    {(sysname[,sysgrpname,sysname...])             }   {name  }
                                                    {name-a}
```

SEND Command (SE)

Purpose: Use the SEND command to communicate with other operators in a multiple-console support (MCS) environment. In a time-sharing environment use the SEND command to communicate with a specific terminal user or all terminal users, and to manage the SYS1.BROADCAST data set.

Example: The following SEND command communicates messages to other operators at MCS consoles.

```
SE {'message'},{BRDCST      }
  {msgno   } {OPERATOR=routecode}
              {CN=console   }
```

Example: The following SEND command communicates with specific time-sharing users.

```
SE {'message'},USER=(userid[,userid]...),{NOW|LOGON},{WAIT|NOWAIT}
  {msgno   }
```

Example: The following SEND command sends a message to all terminal users currently logged on the system.

SEND Command

```
SE {'message'}, {NOW|LOGON}, {WAIT|NOWAIT}, {ROUTE={*ALL|systemname|groupname}}
{msgno }
```

Example: The following SEND command stores messages in the broadcast data set.

```
SE {'message'}, {USER=(userid[,userid]...)}, SAVE
{ msgno } {ALL }
```

Example: The following SEND command lists the requested message (or all messages if that operand is omitted) in the notices section of the broadcast data set.

```
SE msgno,DELETE
```

Example: The following SEND command deletes the specified message from the broadcast data set.

```
SE [msgno,]LIST
```

SET Command (T)

Purpose: Use the SET command to initialize the local time and date and many of the system parameters in SYS1.PARMLIB. Use this command only on direction of the system programmer.

SET or T

```
T [[DATE=yyyy.ddd] [,CLOCK=hh.mm.ss]] [RESET]
[,IPS=xx] [,OPT=xx] [,ICS=xx] [,SMF=xx] [,DAE=xx]
[,MPF={ (xx[,xx]...) |NO}]
[,SLIP=xx] [,PFK=xx] [,IOS=xx] [,EXS=xx]
[,SMS=xx[,RLS_MAXCFFEATURELEVEL({A|Z})]]
[,MMS={xx|NO}]
[,PROG=(xx[,xx...])]
[,DIAG=xx] [,GRSRNL=(xx[,xx]...)]
[,APPC=(xx[,xx]...,L)] [,ASCH=(xx[,xx]...,L)] [,SCH=(xx[,xx]...,L)]
[,CNGRP={ (xx[,xx]...) } |NO ] [,PROD=(xx[,xx]...)]
[,OMVS=(xx[,yy...,nn])]
[,RTLS=(xx[,xx...])]
[,IKJTSO=xx]
```

Note: You may specify the operands in any order, and must specify at least one operand. Do not put a comma before the first operand you specify. If you specify DATE or RESET in a position other than the first, be sure to precede it with a comma. If you specify only one parmlib member with APPC=, ASCH=, SCH=, GRSRNL=, CNGRP=, MPF=, OMVS=, or RTLS=, you do not need to enter the parentheses.

SETDMN Command

SETDMN Command (SD)

Purpose: Use the SETDMN command to change existing values of parameters in a single domain. Issue it only on direction of the system programmer. It is not valid on systems operating in workload management goal mode.

SETDMN or SD
<pre>SD domainnum, {MIN=n1 [,MAX=n2] [,ASRV=(n0,n9)] } ,DSRV=(n0,n9) ,FIXCIDX=nnn { {ASRV=(n0,n9) } [,MIN=n1] [,MAX=n2] } { {DSRV=(n0,n9) } { {FIXCIDX=nnn } {MAX=n2 [,ASRV=(n0,n9)] [,MIN=n1] } ,DSRV=(n0,n9) ,FIXCIDX=nnn</pre>

SETETR Command

Purpose: Use the SETETR command to enable external time reference (ETR) ports that have been disabled.

SETETR
<pre>SETETR PORT=n</pre>

SETGRS Command

Purpose: Use the SETGRS command to migrate a currently active GRS (global resource serialization) ring complex to a GRS star complex or to modify the current RESMIL or TOLINT values.

SETGRS
<pre>SETGRS {MODE=STAR { [RESMIL=nnnnn] [,TOLINT=nnnnn] [,SYNCHRES={YES NO}] } RESMIL=OFF</pre>

SETIOS Command

Purpose: Use the SETIOS command to add, delete, modify, or replace any previously-specified missing interruption handler (MIH) or input/output timing (IOT) parameter dynamically, or to enable or disable the dynamic channel path management function.

SETIOS
<pre>SETIOS DCM={YES NO REFRESH} MIH[,class=mm:ss[,class=mm:ss]...] [,MOUNTMSG={YES NO}] [,DEV=({[/]devnum[,[/]devnum]...),TIME=mm:ss,IOTIMING=mm:ss} {[/]lowdevnum-[/]highdevnum)}] [,MSGONLY={YES NO}]</pre>

SETLOAD Command

Purpose: Use the SETLOAD command to switch dynamically from one parmlib concatenation (logical parmlib) to another without having to initiate an IPL.

SETLOAD
<pre>SETLOAD xx,PARMLIB[, {DSNAME DSN}=dsn] [, {VOLUME VOL VOLSER}=vol]</pre>

SETLOGRC Command

Purpose: Use the SETLOGRC command to change the logrec error and environmental recording medium.

SETLOGRC
<pre>SETLOGRC {LOGSTREAM DATASET IGNORE}</pre>

SETOMVS Command

Purpose: Use the SETOMVS command to change the options dynamically that z/OS UNIX System Services uses. These options are originally set in the BPXPRMxx member of SYS1.PARMLIB at the time of initially program loading (IPL'ing) the system.

SETOMVS Command

SETOMVS	SETOMVS EXTENSIONS (sysplex exclusive)
SETOMVS [AUTHPGMLIST='authprogramlist' NONE] [,FORKCOPY=(COPY COW)] [,IPCSEMNIDS=ipcsemnids] [,IPCSEMNOPS=ipcsemnops] [,IPCSEMNSEMS=ipcsemnsems] [,IPCMSGQBYTES=ipcmsgqbytes] [,IPCMSGNIDS=ipcmsgnids] [,IPCshmPAGES=ipcshmpages] [,IPCshmNIDS=ipcshmnids] [,IPCshmNSEGS=ipcshmnsegs] [,IPCshmSPAGES=ipcshmspapes] [,IPCMSGQMNUM=ipcmsgqmnum] [,LIMMSG=[NONE SYSTEM ALL]] [,MAXASSIZE=maxassize] [,MAXCORESIZE=maxcoresize] [,MAXCPUTIME=maxcputime] [,MAXFILEPROC=maxfileproc] [,MAXFILESIZE=(maxfilesize NOLIMIT)] [,MAXMMAPAREA=maxmmaparea] [,MAXPROCSYS=maxprocsys] [,MAXPROCUSER=maxprocuser] [,MAXPTYs=maxpty] [,MAXSHAREPAGES=maxsharepages] [,MAXTHREADS=maxthreads] [,MAXTHREADTASKS=maxthreadtasks] [,MAXUIDS=maxuids] [,PID=pid,processlimitname=newvalue] [,PRIORITYGOAL=(n) NONE] [,PRIORITYPG=(n) NONE]; [,RESET=(xx)] [,STEPLIBLIST='stepliblist'] [,SUPERUSER=superuser] [,SYNTAXCHECK=(xx)] [,TTYGROUP=ttygroup] [,USERIDALIASTABLE='useridaliastable'] [,VERSION='string']	SETOMVS FILESYS ,FILESYSTEM=filesystem ,AUTOMOVE=YES NO UNMOUNT indicator(sysname1 ,sysname2,...,sysnameN) or SETOMVS FILESYS ,FILESYSTEM=filesystem ,SYSNAME=sysname * or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,AUTOMOVE=YES NO UNMOUNT indicator(sysname1 ,sysname2,...,sysnameN) or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,AUTOMOVE=YES NO UNMOUNT indicator(sysname1 ,sysname2,...,sysnameN) or SETOMVS FILESYS ,MOUNTPOINT=mountpoint ,SYSNAME=sysname * or SETOMVS FILESYS ,FROMSYS=sysname ,SYSNAME=sysname * Note: FILESYSTEM, FROMSYS, and MOUNTPOINT are mutually exclusive parameters. When you specify FILESYS, you must supply one of these three parameters.

SETPROG Command

Purpose: Use the SETPROG command to update the authorized program facility (APF) list, dynamic exits, and LNKLIST concatenations, and to manage the LPA content.

Example: The following SETPROG command updates the APF list.

```
SETPROG APF{,FORMAT={DYNAMIC|STATIC}}
        {,{ADD|DELETE},DSNAME|LIBRARY=libname,{SMS|VOLUME=volume} }
```

Example: The following SETPROG command updates dynamic exits.

SETPROG Command

```

SETPROG EXIT, {ADD, EXITNAME=exitname, MODNAME=modname      }
                [, STATE={ACTIVE|INACTIVE}]
                [, DSNAME=dsname]
                [, JOBNAME={jobname|*}]
                [, ABENDNUM=(n[, CONSEC])]
                [, FIRST|LAST]

                {ATTRIB, EXITNAME=exitname, KEEPRC=(compare, kk)}

                {DELETE, EXITNAME=exitname, MODNAME=modname  }
                [, FORCE={YES|NO}]

                {MODIFY, EXITNAME=exitname, MODNAME=modname  }
                [, STATE={ACTIVE|INACTIVE}]
                [, JOBNAME={jobname|*}]

                {UNDEFINE, EXITNAME=exitname                  }

```

Example: The following SETPROG command updates linklist concatenations by:

- Defining a LNKLST set of data sets for the LNKLST concatenation
- Adding data sets to or deleting data sets from the LNKLST set
- Removing the definition of a LNKLST set from the system
- Testing for the location of a specific module in the LNKLST concatenation
- Activating a LNKLST set as the LNKLST concatenation for the system
- Updating an address space for jobs to use a LNKLST set.

```

SETPROG LNKLST, {DEFINE, NAME=lnklstname[, COPYFROM=lnklstname] [, NOCHECK] }
                {ADD, NAME=lnklstname,
                  DSNAME=dsname[, VOLUME=volser] [, ATBOTTOM      ]
                  [, ATTOP          ]
                  [, AFTER=dsname]
                  [, CONCAT(CHECK | NOCHECK)]
                }
                {DELETE, NAME=lnklstname, DSNAME=dsname }
                {UNDEFINE, NAME=lnklstname }
                {TEST, NAME=lnklstname, MODNAME=name }
                {ACTIVATE, NAME=lnklstname }
                {UPDATE, {JOB=jobname} }
                  {ASID=asid }
                {UNALLOCATE }
                {ALLOCATE }

```

Example: The following SETPROG command specifies, any time after IPL, modules to add to or delete from the LPA, and the minimum amounts of CSA storage that must remain available after an ADD operation.

```

SETPROG LPA, {ADD, [MODNAME=(modname..., modname) | MASK=mask]
              , DSNAME=[dsname | LNKLST]
              [, FIXED] [, PAGEPROTPAGE]

              {DELETE, MODNAME=(modname..., modname) }
              FORCE=YES [CURRENT | OLDEST]

              {CSAMIN=(below, above) }

```

SETRRS Command

SETRRS CANCEL Command

Purpose: Use the SETRRS CANCEL command to cancel (abnormally terminate) resource recovery services (RRS).

SETRRS
SETRRS CANCEL[,DUMP NODUMP]

SETSMF Command (SS)

Purpose: Use the SETSMF command to add a SUBPARM parameter or replace any previously-specified parameter in the active SMF member of SYS1.PARMLIB except the ACTIVE, PROMPT, SID, or EXITS parameters. Use the abbreviation SS to contrast this SETSMF command with the SET SMF command.

SETSMF or SS
SS parameter(value[,value]...)

SETSMS Command

Purpose: Use the SETSMS command to change a subset of storage management subsystem (SMS) parameters from the console without changing the active IGDSMSxx member of SYS1.PARMLIB. Contrast this SETSMS command with the SET SMS command.

SETSMS
SETSMS parameter(value)[,parameter(value)]...

SETSSI Command

Purpose: Use the SETSSI command to add, activate, or deactivate a subsystem dynamically.

SETSSI
SETSSI {ADD,{SUBNAME SUB S}=subname [, {CONSNAME C}=consname] [, {INITRTN I}=initrtn[, {INITPARM P}=initparm]} {DEACTIVATE DEACT},{SUBNAME SUB S}=subname } {ACTIVATE ACT},{SUBNAME SUB S}=subname }

SETXCF Command

Purpose: Use the SETXCF command to control the cross-system coupling facility (XCF).

Example: The following SETXCF COUPLE command can switch the current alternate XCF couple data set to the primary XCF couple data set, or specify an alternate XCF couple data set, or change options specified in the COUPLExx parmlib member.

```

SETXCF COUPLE, {PSWITCH }
                {ACOUPL= (alternatedsname [, alternatevolume]) }
                {INTERVAL= timeinterval }
                {OPNOTIFY= timeinterval }
                {CLEANUP= timeinterval }
                {MAXMSG= defaultmaxmsgbuffers }
                {RETRY= defaultretrylimit }
                {CLASSLEN= defaultclasslength }
                {TYPE= (name [, name] ...), }
                    {PCOUPLE= (primarydsname [, primaryvolume]) }
                    {ACOUPL= (alternatedsname [, alternatevolume]) }
                    {PSWITCH }

```

Example: The following SETXCF FORCE command cleans up resources related to structures in a coupling facility. The resources can be either structures actively in use in the sysplex, or dumps associated with structures pending deallocation.

```

SETXCF FORCE,
  {STRUCTURE, STRNAME= (strname [, strname] ... ) }
  {CONNECTION, STRNAME= strname, CONNAME= { (conname [, conname] ...) | ALL } }
  {STRDUMP, STRNAME= strname [, STRDUMPID= strdumpid] }
  {STRDUMPSERIAL, STRNAME= strname [, STRDUMPID= strdumpid] }

```

Example: The following SETXCF MODIFY command changes current XCF parameters.

SETXCF Command

```
SETXCF MODIFY, {PATHIN, {DEVICE=([/]indevnum[, [/]indevnum]...)} }
                {STRNAME=(strname[, strname]...)} }
                [,MAXMSG=maxmsgbuffers]
                [,RETRY=retrylimit]

                {PATHOUT, {DEVICE=([/]outdevnum[, [/]outdevnum]...)} }
                {STRNAME=(strname[, strname]...)} }
                [,CLASS=classname]
                [,MAXMSG=maxmsgbuffers]
                [,RETRY=retrylimit]

                {LOCALMSG, MAXMSG=maxmsgbuffers }
                [,CLASS=class-name]

                {CLASSDEF, CLASS=classname }
                [,CLASSLEN=classlength]
                [,MAXMSG=defaultmaxmsgbuffers]
                [,ADDGROUP=(groupname[, groupname]...)]
                [,DELGROUP=(groupname[, groupname]...)]
```

Example: The following SETXCF PRSMPOLICY (or PRSMPOL) command either activates an XCF PR/SM policy or deactivates a currently active XCF PR/SM policy.

```
SETXCF PRSMPOLICY, {DEACTIVATE|ACTIVATE=memname}
```

Example: The following SETXCF START command starts signalling paths and defines transport class definitions for use by the XCF.

```
SETXCF START, {CLASSDEF, CLASS=classname }
                [,CLASSLEN=classlength]
                [,MAXMSG=maxmsgbuffers]
                [,GROUP=(groupname[, groupname]...)]

                {PATHIN, {DEVICE=([/]indevnum[, [/]indevnum]...)} }
                {STRNAME=(strname[, strname]...)} }
                [,MAXMSG=maxmsgbuffers]
                [,RETRY=retrylimit]

                {PATHOUT, {DEVICE=([/]outdevnum[, [/]outdevnum]...)} }
                {STRNAME=(strname[, strname]...)} }
                [,MAXMSG=maxmsgbuffers]
                [,RETRY=retrylimit]
                [,CLASS=classname]

                {POLICY, TYPE=name, POLNAME=polname }

                {REBUILD, {POPULATECF=cfname} }
                {DUPLEX, }
                {STRNAME=(strname[, strname]...)}
                {CFNAME=(cfname[, cfname]...)} }
                [,LOCATION={NORMAL|OTHER}]
                [,LESSCONN={TERMINATE|CONTINUE}]

                {ALTER, STRNAME=strname, SIZE=size } }
```

Example: The following SETXCF STOP command removes XCF signalling paths and transport class definitions.

```

SETXCF STOP, {PATHIN, {DEVICE=([/]indevnum[, [/]indevnum]...)} }
                {STRNAME=(strname[, strname]...)} }
                [, UNCOND=NO|YES]

                {PATHOUT, {DEVICE=([/]outdevnum[, [/]outdevnum]...)} }
                {STRNAME=(strname[, strname]...)} }
                [, UNCOND=NO|YES]

                {CLASSDEF, CLASS=classname }

                {POLICY, TYPE=name }

                {REBUILD, {POPULATECF=cfname} }
                {DUPLEX, }
                {STRNAME=(strname[, strname]...)} }
                {, KEEP=NEW|OLD}
                {CFNAME=(cfname[, cfname]...)} }

                {ALTER, STRNAME=strname }
    
```

SLIP Command (SL)

Purpose: Use the SLIP command to control serviceability level indication processing (SLIP), a diagnostic aid that intercepts or traps certain system events. The three types of the SLIP command set, modify, and delete SLIP traps. This table shows an overview of the SLIP command syntax.

SLIP or SL	
SLIP SET[,options],END	Command for an error event trap (non-PER)
SLIP SET,IF[,options],END	Command for an instruction fetch PER trap
SLIP SET,SBT[,options],END	Command for a successful branch PER trap
SLIP SET,SA SAS[,options],END	Commands for a storage alteration PER trap
SLIP MOD[,options]	Command to modify an existing trap
SLIP DEL[,options]	Command to delete an existing trap

Notes:

- You must specify SET, MOD, or DEL immediately following SLIP.
- If you specify IF, SBT, SA, or SAS, it must immediately follow SET.
- You must specify END at the end of all SLIP SET commands.

Examples: The following tables show the syntax for various forms of the SLIP SET command.

SLIP Command

```
SLIP SET

SLIP SET

[,ADDRESS=(start[,end])
  [,LPAEP=(name[,start[,end]])
  [,LPAMOD=(name[,start[,end]])
  [,NUCEP=(name[,start[,end]])
  [,NUCMOD=(name[,start[,end]])
  [,PVTEP=(name[,start[,end]])
  [,PVTMOD=(name[,start[,end]])

[,ASID=(asid[,asid]...) ]
[,COMP=code[,REASON=code] ]
[,DATA=(comparison[,comparison]...) ]
[,ERRTYP=(type,[,type]...) ]
[,JOBNAME={userid | jobname} ]
[,JSPGM=name ]
[,MSGID=message-id ]
[,MODE=(mode[,mode]...[,ANY | EVERY]) ]
[,PSWASC=(mode[,mode]...) ]

[,ACTION=[IGNORE[,option]] ]
          [(nodump[,nodump]...) [,option] ]
          [NOSUP[,option] ]
          [RECORD ]
          [SVCD[,options] ]
          [TRACE[,options] ]
          [TRDUMP[,options] ]
          [STOPGTF[,options] ]
          [WAIT[,options] ]

[,ENABLE | ,DISABLE]

[,IDGROUP=idgroup]

[,MATCHLIM=m ]
  [,MATCHLIM=1 for ACTION=SVCD or ACTION=SYNCSVCD

[,DEBUG]

[,ID=trapid]

[,OK]

[,RBLEVEL={ERROR | NOTSVRB | PREVIOUS}]

,END
```


SLIP SET,IF or SET,SBT

```
SLIP SET,{IF|SBT}
```

```
[,LPAEP=(name[,start[,end]]) ]
[,LPAMOD=(name[,start[,end]])
[,NUCEP=(name[,start[,end]])
[,NUCMOD=(name[,start[,end]]
[,PVTEP=(name[,start[,end]])
[,PVTMOD=(name[,start[,end]])
[,RANGE=(start[,end])

[,ASID=(asid[,asid]...)]

[,DATA=(comparison[,comparison]...)]

[,JOBNAME={userid | jobname}]

[,JSPGM=name]

[,MODE= (mode[,mode]...[,ANY | EVERY])]

[,PSWASC=(mode[,mode]...)]

[,ACTION=
    [(IGNORE[,RECOVERY])] ]
    [(RECOVERY[,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options] ]
    [(STDUMP[,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options] ]
    [(STRACE[,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options] ]
    [(SVCD[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options] ]
    [(SYNCSVCD[,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options] ]
    [(STOPGTF[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options] ]
    [(TRACE[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options] ]
    [(TRDUMP[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options] ]
    [(WAIT[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options] ]

[,ENABLE | ,DISABLE]

[,IDGROUP=idgroup]

[,MATCHLIM=m ]
    [,MATCHLIM=1 for ACTION=SVCD or ACTION=SYNCSVCD
    [,MATCHLIM=50 for ACTION=STDUMP or ACTION=STRACE

[,PRCNTLIM=p | ,PRCNTLIM=10]

[,DEBUG]

[,ID=trapid]

[,OK]

,END
```

SLIP Command

SLIP SET,SA or SET,SAS
<pre> SLIP SET,{SA SAS} [,ADDRESS=(start[,end]) [,LPAEP=(name[,start[,end]]) [,LPAMOD=(name[,start[,end]]) [,NUCEP=(name[,start[,end]]) [,NUCMOD=(name[,start[,end]]) [,PVTEP=(name[,start[,end]]) [,PVTMOD=(name[,start[,end]]) [,RANGE=(start[,end])] [,ASID=(asid[,asid]...)] [,ASIDSA=(asid 'jobname'[,asid 'jobname']...)] [,DATA=(comparison[,comparison]...)] [,DSSA=(asid.name 'jobname'.name[,asid.name 'jobname'.name]...)] [,JOBNAME={userid jobname}] [,JSPGM=name] [,MODE=(mode[,mode]...[,ANY EVERY])] [,PSWASC=(mode[,mode]...)] [,ACTION= [(IGNORE[,RECOVERY])] [(RECOVERY[,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options]] [(STDUMP[,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options]] [(STOPGTF[,REFAFTER] [,REFBEFOR] [,TARGETID])[,options]] [(STRACE[,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options]] [(SVCD[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options]] [(SYNCSVCD[,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options]] [(TRACE[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options]] [(TRDUMP[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options]] [(WAIT[,RECOVERY] [,REFAFTER] [,REFBEFOR] [,STOPGTF] [,TARGETID])[,options]] [,ENABLE ,DISABLE] [,IDGROUP=idgroup] [,MATCHLIM=m] [,MATCHLIM=1 for ACTION=SVCD or ACTION=SYNCSVCD [,MATCHLIM=50 for ACTION=STDUMP or ACTION=STRACE [,PRCNTLIM=p ,PRCNTLIM=10] [,DEBUG] [,ID=trapid] [,OK] ,END </pre>

Example: The following SLIP command modifies a SLIP trap.

SLIP MOD
<pre> SLIP MOD{,ENABLE ,DISABLE}{,ALL ,ID=trapid} </pre>

Example: The following SLIP command deletes one or all SLIP traps.

SLIP DEL

```
SLIP DEL{,ALL | ,ID=trapid}
```

START Command (S)

Purpose: Use the START command to start started tasks, which support system functions such as IMS, CICS, and RACF. Started tasks are defined in cataloged procedures (residing in procedure libraries) or in jobs residing in a partitioned data set defined in master JCL.

Notes:

- For any variation of the START command, if the subsystem that processes the task is JES2 or JES3 and you omit *devicetype* (or *devnum*), or *classes*, or *volumeserial*, you must supply a comma for each one of these parameters that you leave out. Do not enter any commas, however, after the last parameter that you specify.
- *devicetype* and *devnum* are mutually exclusive. You cannot specify both parameters on a START command.
- *JOBNAME* and *identifier* are mutually exclusive. You cannot specify both parameters on a START command.
- If you specify an option in apostrophes, use uppercase.
- If you are overriding a data set name that is 44 characters long, use *DSN=*, not *DSNAME=*.
- If you are overriding a symbolic parameter, do not use any DD statement keyword parameters.

Example: The following START command starts a system task from the console.

```
S membername[.identifier][,devicetype|,[/]devnum][,volumeserial]
  [,parameters][,JOBNAME=jobname][,JOBACCT=acct_info]
  [,SUB=subsystemname][,keyword=option[,keyword=option]...]
```

Example: The following START command starts the advanced program-to-program communication/MVS (APPC/MVS) address space.

```
S APPC,SUB=MSTR[,APPC=(nn[,nn]...[,L])]
```

Example: The following START command starts the APPC/MVS transaction scheduler (ASCH) address space.

```
S ASCH,SUB=MSTR[,ASCH=(nn[,nn]...[,L])]
```

Example: The following START command starts the generalized trace facility (GTF).

```
S {GTF|membername}[.identifier][,devicetype|,[/]devnum][,volumeserial]
  ([,MODE={INT|DEFER|EXT}][,TIME=YES][,DEBUG=YES])
  [,BLOK={numpages|nnk|nnM}][,MEMBER=xxxxxxx][,REGION=nnnnk]
  [, {SADMP|SA}={nnM|nnk}][, {SDUMP|SD}={nnM|nnk}]
  [, {NOPROMPT|NP}][, {ABDUMP|AB}={nnM|nnk}]
  [,keyword=option[,keyword=option]...]
```

START Command

Example: The following START command starts the library lookaside (LLA) address space.

```
S LLA[,SUB=MSTR][,LLA=xx]
```

Example: The following START command starts the object access method.

```
S {OAM[membername]}.identifier[,OAM=xx]
```

Example: The following START command starts resource recovery services.

```
S RRS|membername[,CTMEM=CTnRRSxx][,GNAME=|grpname][,JOBNAME=jobname]
```

Example: The following START command starts TSO/VTAM time-sharing.

```
S membername|.identifier[,devicetype|,[/]devnum]
  [,volumeserial][([,MEMBER={name|nn})[,USERMAX=nnnnn]
  [,GNAME=[name|NONE]])
  [,keyword=option[,keyword=option]...]
```

Example: The following START command starts the virtual lookaside facility (VLF) or the data lookaside facility (DLF).

```
S {VLF|DLF},SUB=MSTR[,NN=xx]
```

Example: The following START command starts an external writer.

```
S {XWTR|membername}.identifier[,devicetype|,[/]devnum]
  [,volumeserial][,classes][,keyword=option[,keyword=option]...]
```

STOP Command (P)

Purpose: Use the STOP command to stop system functions and jobs in execution.

Example: The following STOP command stops a started task.

STOP Command

```
P [jobname.]identifier[,A=asid]
```

Example: The following STOP command stops the APPC/MVS transaction scheduler (ASCH) address space.

```
P {ASCHINT,A=asid}
```

Example: The following STOP command stops the data lookaside facility (DLF) address space.

```
P DLF
```

Example: The following STOP command stops the library lookaside facility (LLA) address space.

```
P LLA
```

Example: The following STOP command stops the object access method (OAM).

```
P [OAM.|jobname.]identifier
```

Example: The following STOP command stops the virtual lookaside facility (VLF).

```
P VLF
```

STOPMN Command (PM)

Purpose: Use the STOPMN command to stop displays initiated in response to the MONITOR command or the MONITOR parameters on the CONSOLE and INIT statements in the CONSOLxx member of SYS1.PARMLIB.

STOPMN or PM

```
PM {JOBNAMES[,L={a|cc|cca|name|name-a}]}  
  {DSNAME }  
  {SPACE }  
  {STATUS[,L={a|cc|cca|name|name-a}]}  
  {SESS[,L={a|cc|cca|name|name-a}] }
```


TRACE	
TRACE	[STATUS]
	[ST[,nnnK][,BR={ON OFF}]]
	[ST[,OFF]]
	[MT[,nnnK ,OFF]]
	[CT{,WTRSTART=membername[,WRAP NOWRAP] }]
	[CT{,WTRSTOP=jobname}]
	[CT{[,ON] ,COMP=name[,SUB=(subname)] [,PARM=mem] }]
	,nnnnK
	,nnnnM
	,OFF
	[TT{[,COLL C=collection name }]
	{,CON=connection type }]
	{,COR=correlation info }]
	{,LU=logical unit name }]
	{,LVL=level }]
	{,NET=netid }]
	{,PKG=package name }]
	{,PLAN= PL=plan name }]
	{,PRF=perform }]
	{,PROC PR=proc name }]
	{,PRS=process }]
	{,SUB=subsystem }]
	{,TC=transaction class }]
	{,TRAN T=transaction name}]
	{,USER U=userid }]
	[,WTR=membername STOP]
	[,LATENT=Y N]
	[,BUFSIZ=nnnK nnM]]
	[,OFF={nn ALL}]

TRACK Command (TR)

Purpose: Use the TRACK command to request a periodic display of job information on display consoles.

TRACK or TR
TR {TS JOBS J A} [,LIST ,L] [,USERID=userid] [,L={a cc cca name name-a}]

Note: To stop the TRACK display, use the STOPTR command.

UNLOAD Command (U)

Purpose: Use the UNLOAD command to unload mounted tape or DASD volumes, where *devnum* is the device number of the I/O device containing the volume to unload.

UNLOAD or U
U [/]devnum

VARY Command

VARY Command (V)

Purpose: Use the VARY command to control certain system components such as the online or offline status of consoles and I/O devices.

Example: The following VARY command places the system console in problem determination mode.

```
VARY CN(conspec1|*),{ACTIVATE|ACT}
```

Example: The following VARY command removes the system console from problem determination mode.

```
VARY CN(conspec1|*),{DEACTIVATE|DEACT}
```

Example: The following VARY command changes the indicated authority for the console id(s) or name(s) specified.

```
V CN{(*|conspec1[,conspec1]... )}
  [,ALTGRP={name|*NONE*}]
  [,AMSCOPE=([*][,name[,name]...)]
  [,AUTH={ALL|INFO|MASTER}([SYS][,IO][,CONS])}]
  [,AROUT=(rtcode[,rtcode]...)]
  [,DMSCOPE=([*][,name[,name]...)]
  [,DROUT=(rtcode[,rtcode]...)]
  [,LOGON={OPTIONAL|REQUIRED|AUTO|DEFAULT}]
  [,LU={luname|*NONE*}]
  [,MSCOPE=(*ALL)|{([*][,name[,name]...)]}]
  [,OFFLINE|,ONLINE[,SYSTEM=sysname][,FORCE]]
  [,ROUT={ALL|NONE}(rtcode[,rtcode]...)}]
  [,UD={Y|N}]
```

Example: The following VARY command places a secondary console online or offline.

```
V (conspec[,conspec]...),{OFFLINE|ONLINE}
```

Example: The following VARY command assigns and controls multiple console support (MCS) consoles.

```
V {conspec2|(conspec2[,conspec2]...),CONSOLE
  [,ALTCONS=conspec2]
  [,AROUT=(rtcode[,rtcode]...)]
  [,AUTH={ALL|MASTER|INFO}([SYS][,IO][,CONS])}]
  [,DROUT=(rtcode[,rtcode]...)]
  [,ROUT={ALL|NONE}(rtcode[,rtcode]...)}]
```


Example: The following VARY command switches the master console.

```
V {[/]devnum},MSTCONS
 {conname }
```

Example: The following VARY command controls the hardcopy message set and the hardcopy medium.

```
V [[/]devnum],HARDCPY[,CMDS|,NOCMDS|,STCMDS|,INCMDS]
 |conname          [ ,AROUT=(rtcode[,rtcode]...) ]
 |SYSLOG           [ ,DROUT=(rtcode[,rtcode]...) ]
 |OPERLOG          [ ,ROUT={ALL|NONE } ]
                   { (rtcode[,rtcode]...) }
                   [ ,UD={Y|N} ]
                   [ ,OFF[,UNCOND]]
```

Example: The following VARY command turns on or off the AUTOSWITCH attribute of a tape device.

```
V {(devspec[,devspec]...)}, { [AUTOSWITCH|AS] [,ON|OFF] }
```

Example: The following VARY command places an I/O device or range of devices online or offline.

```
V {(devspec[,devspec]...)}, { ONLINE[,UNCOND] [,FORCE] }
 {devspec } | ,SHR
 | ,RESET
 {OFFLINE[,FORCE] }
```

Example: The following VARY command controls a global resource serialization (GRS) ring complex.

```
V GRS{({sysname|*|ALL}), {RESTART|R}}
 {({sysname|*}), {QUIESCE|Q} }
 {(sysname), {PURGE|P} }
```

Example: The following VARY command places an I/O path or paths online or offline.

VARY Command

```
V PATH
  {[/]devnum,chp[,[/]devnum,chp]...}
  {([/]devnum[,[/]devnum]...),chp
   [,([/]devnum[,[/]devnum]...),chp]...}
  {([/]lowdevnum-[/]highdevnum[,[/]lowdevnum-[/]highdevnum]...),chp
   [,([/]lowdevnum-[/]highdevnum[,chp]...)}
  {(cfname,chp[,cfname,chp]...)}
  {((cfname[,cfname]...),chp
   [, (cfname[,cfname]...),chp]...)}
  {,ONLINE[,FORCE]}
  {,OFFLINE[,UNCOND | ,FORCE]}
```

Example: The following VARY command changes the state of coupling facility cache structures and volumes over a sysplex wide scope.

```
V SMS,{CFCACHE(cachename),{ENABLE|E }
      {QUIESCE|Q}
      {CFVOL(volid),{ENABLE|E }
      {QUIESCE|Q}
      {MONDS(dsname[,dsname...]),{ON|OFF}
      {SHCDS(shcdsname),{NEW }
      {NEWSPARE}
      {DELETE }
      {SMSVSAM,{ACTIVE
      {SPHERE(spherename),{ENABLE|E}}
      {FALLBACK
      {TERMINATESERVER
      {FORCEDELETELOCKSTRUCTURE }
```

Example: The following VARY command places an optical drive or optical library online or offline.

```
V SMS,{DRIVE|DRI|LIBRARY|LIB}(name),{ONLINE|ON }
      {OFFLINE|OFF}
```

Example: The following VARY command places a system-managed tape library online or offline.

```
V SMS,{LIBRARY}(name[, {systemid[,systemid]...}]),{ONLINE|ON }
      {LIB } {ALL } {OFFLINE|OFF}
      {*_ }
```

Example: The following VARY command changes the SMS status of a storage group.

VARY Command

```
V SMS,{{STORGRP|SG}(storgrp,[*|ALL|system[,system]...)}},{QUIESCE|Q}{[,NEW|,N]
      {
      {VOLUME|VOL}(volume,[*|ALL|system[,system]...)} {ENABLE|E }
      {DISABLE|D}{[,NEW|,N]
```

Example: The following VARY command controls an application environment.

```
V WLM,APPLENV=applenvname,{REFRESH }
                          {QUIESCE|Q}
                          {RESUME }
```

Example: The following VARY command activates a workload management service policy for a sysplex.

```
V WLM,POLICY=policyname[,REFRESH]
```

Example: The following VARY command causes the cross-system coupling facility (XCF) to remove the specified system from the sysplex.

```
V XCF,systemname,{OFFLINE|OFF}{[,RETAIN={YES|NO}]}[,FORCE]
```

WRITELOG Command (W)

Purpose: Use the WRITELOG command to control the system log.

```
WRITELOG or W
W [class|CLOSE|START]
```

WRITELOG Command

Appendix. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen-readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen-readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using it to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Volume I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

Notices

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

NOTICES

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States and/or other countries:

- CICS
- DFSMS/MVS
- IBM
- IBMLink
- IMS
- MVS
- OS/390
- PR/SM
- RACF
- Resource Link
- VTAM
- zSeries
- z/OS
- z/OS.e

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Readers' Comments — We'd Like to Hear from You

z/OS
MVS System Commands Summary

Publication No. SA22-7628-03

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5694-A01, 5655-G52

Printed in U.S.A.

SA22-7628-03

